

Dynamic Vector and Raster Integrated Data Model Based on Code-Points

Min Sun¹, Yong Xue^{2,3}, Ai-Nai Ma¹ and Shan-Jun Mao¹

¹The Institute of GIS & RS of Peking University, Beijing, 100871 China
{Sm6902@263.net, maainai@pku.edu.cn}

²Laboratory of Remote Sensing Information Sciences, Institute of Remote Sensing
Applications, Chinese Academy of Sciences, Beijing, 100101, China
{yxue@irsa.irsa.ac.cn}

³School of Informatics and Multimedia Technology, University of North London, 166-220
Holloway Road, London N7 8DB, UK
{y.xue@unl.ac.uk}

Abstract. With the rapid development of remote sensing technology, the integration of raster data and vector data becomes more and more important. Raster data models are used in tessellation spaces and vector data models are used in discrete spaces respectively. The relationships between tessellation space and discrete space have to be established for integrated data models. The minimum cells where both raster and vector data could be processed have to be defined. As it is very easy to establish relationships between vector points and corresponding raster cells, we defined those raster cells as Code-Points, the minimum cells where both raster and vector data could be processed. All vector elements such as lines, faces and bodies are composed directly or indirectly of Code-Points. This can be done by using interpolation algorithms to Code-Points in real-time. We have developed an integrated data model based on above procedures. In addition, we have developed a geometric primitive library for 3-Dimensional objects in order to improve the processing speed. This library could be hardware realized as a graphic accelerator card. If the conversion between vector and raster could be done in real time, the integrated data model could be used for operational integration of remote sensing and GIS.

1. Introduction

The main purpose of remote sensing, or Earth observation, is to provide information about the surface of the Earth. The launches of high-resolution satellites have effectively promoted the development of remote sensing technology. High-resolution satellite images increasingly become the main data source for Geographic Information Systems (GISs). The integration of GIS and remote sensing becomes more important. The fundament of the integration of remote sensing and GIS is the integrated vector and raster data models. Many related research works in this field have been done, e.g.: Ballard [1] developed a Strip Tree Model, dividing vector lines into several segments, and expressed vector lines using binary tree data structure; Randal [6] put forward to

express vector data using quadtree hierarchical structure; Gong [2] developed a vector and raster integrated data model using multilevel grid and quadtree data structure.

The real world is very complex and it is very difficult to build a common 3D data model. Molannar [4] developed a 3DFS data model. Pilouk *et al.* [5] developed a tetrahedral based data model. Li [3] developed a CSG based data model and Tempfle [8] developed a 3D topological data model for 3D city. All these data models are vector data models.

If vector data of a face element is expressed by quadtree, the quadtree expressed data layer must be repartitioned in quadtree when vector data changed. Because of such defects, there are still no operational GISs which could integrate vector and raster data dynamically. The fundamental problem of the fully integrated system is to choose appropriate data structures, those problems are remained unsolved [9].

In the paper, we developed a new method, which is based on the code-point, a minimum unit which both raster data and vector data can be processed directly. First, we discuss its main idea, followed by spatial coding algorithm. Integrated data model is explained in Section 4. After the discussion of data structure in Section 5, the conclusion has been given.

2. Basic Idea of the Integrated Data Model

As we know, a vector object is expressed using discrete points in a discrete space (vector space), and a raster object is expressed using continuous cells in a tessellation space (raster space). In order to integrate vector and raster data, a unified space should be setup and the minimum cells should be identified and defined on which both raster data and vector could be disposed, where two different kinds of data could be processed directly without any conversions.

As discrete space could be considered as a special case of tessellation space, tessellation space could be used for the integration of vector and raster data. Vector point is the basic element in vector space, and raster cell is the minimize cell in raster space. When a raster space is expressed using Octree data structure, raster cell could be expressed using a code. If this code is a vector point, we call it a Code-Point (CP). The Code-Point is the minimum cell in the integrated vector and raster data model developed in the paper. In tessellation space, an object is expressed using a set of raster cells. But for an object in a vector space, it becomes much more difficult as a vector object, especially a solid object, usually combined by lines or faces. In order to represent a vector object in Code-Points, the object must be converted from its present expression to Code-Point expression. Basically, this conversion is same as the conversion from vector to raster. Normally, the conversion of a vector object using raster cells would not only increase the data abundance, but also not be convenient to update data. In our vector and raster integrated data model, we use algorithms to conduct the conversion from vector data to raster data in real-time.

First, raster spaces are represented in octree data structure. Octree is an effective data structure to represent a 3D tessellation space. A vector point corresponds to a raster cell, i.e. an octree node. As it needs a large amount of computer memory to

establish an octree data structure, we use a code-point to represent the relationship between a vector point and a raster cell (an octree node).

Interpolation algorithms will be used to represent vector elements in Code-Points in tessellation space. For example, a line could be expressed by a linear interpolation of Code-Points. However, it will be much more difficult and complex for a face and a solid. In the following section, we developed some methods to solve this problem. Thus, the key point for the integrated data model is the real-time raster interpolation algorithms. Base on the following several reasons, we consider that the real-time interpolation algorithm is feasible:

- Real-time interpolation could avoid data redundancy, and it is easy to update data;
- It could be much more easier to manage and deal the vector objects if the raster form is interpolated from Code-Points in real-time;
- In the interactive graphic mode, the frequent calculation usually occurs in small area. The amount of calculation could be reduced;
- To build a library for primitive geometric objects and related conversion algorithms which could be performed by hardware, would reduce the calculation complexity significantly and improve the processing speed.
- The new developments of data processing capability of the present computer hardware provide good opportunities for the fast calculation in practice.

Figure 1 shows the representation of a line object and a face object based on Code-Points. The raster forms of the line and the face are not continuous. Interpolation process is needed. Firstly, we discuss the algorithm of Space Coding (Octree-ID).

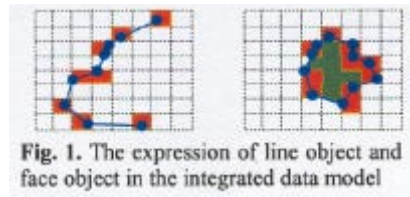


Fig. 1. The expression of line object and face object in the integrated data model

3. Space coding algorithm

The establishment of an octree structure for a raster space would use a large amount of RAM space. In order to save the memory space, we only calculate corresponding raster octree codes for each vector point. Here the code means Morton code of octree node. The main idea of the algorithm is to calculate Morton codes for octree nodes that correspond to the vector points according their coordinates. The algorithm frame is as following:

Algorithm_Calculate_Octree-id (X, Y, Z \rightarrow Octree-ID):

/ Assuming MinX, MaxX, MinY, MaxY, MinZ, MaxZ are the maximum values and the minimum values in x direction, y direction and z direction, respectively. And assuming Min_Dx, Min_Dy and Min_Dz are the least cell values in x direction, y direction and z direction, respectively. The three values could be equal. Then we have */*

$$W_x = \text{MaxX} - \text{MinX};$$

$$W_y = \text{MaxY} - \text{MinY};$$

```

Wz = MaxZ - MinZ;
/* The code in x direction is {0,1}, in y direction needs add {0, 2}, in z direction
needs add {0, 4} as well.*/
/* Define the point structure */
typedef point {
    float x; float y; float z;
    char pCode[10]; }
/* Calculate the minimal interval values in x y and z direction*/
Min_Width = Min( Min_Dx, Min_Dy, Min_Dz);
/* Calculate the length of Octree codes */
t = f(int[Wx / Min_Width] );
/* Calculate Octree-Ids */
for(int i = 0; i < t; dx++)
    for(j = 0; j < pt_num; j++)
        if(p[j].x > (MinX + Wx * (c))) // c = 1/2 + 1/4 + 1/8 + 1/16..... 1/(2^t)
            p[j].pCode[i] = 1;
        else p[j].pCode[i] = 0;
            if(p[j].y > (MinY + Wy * (c)))
                p[j].pCode[i] += 2;
            if(p[j].z > (MinZ + Wz * (c)))
                p[j].pCode[i] += 4;
        }
}

```

The length of Octree-ID is t. The larger t is, the smaller the octree cell is. It also means that the space is partitioned more deeply. The value of t depends on real situations. However, the value can't be too small. Otherwise, it needs much more interpolation calculations. In the next, we discuss the establishment of integrated data model in detail.

4. Integrated data model

4.1 Vector element expression

Normally, vector elements in a vector model are points, lines, faces and bodies. A line is composed of points, a face is composed of lines and a solid is composed of faces. In a raster space, all kinds of elements are expressed in raster cells. Vector points and lines are relatively simple. In the paper, we mainly discuss face elements and solid elements. When a face element is a spatial plan or a combination of a set of spatial plans, it could be expressed by its sidelines. If it is a spatial curve surface, it should be expressed by regular grid or TIN in order to simplify the procession.

Spatial solid objects could be divided into homogeneous objects and heterogeneous objects according to its internality. And it also could be divided into regular objects and irregular objects (here regular objects are spatial objects expressed by regular geometry primitives such as cuboids, prisms, spheres, cylinders and cones. It also

includes complex spatial objects expressed by the combination of these regular geometry primitives. All primitives that could be expressed by parameters could also be called regular geometry primitives). For a homogeneous object, it could be expressed by its surfaces. However, for a heterogeneous object, it can't be simply expressed by its surfaces, as its internal needs to be expressed as well. Molenaar's 3DFS data model is good to express a homogeneous object, and is also good to express a regular object in some degrees [4]. The tetrahedral based data model developed by Pilouk *et al.* [5] is good to express a heterogeneous object, and it is also good to express an irregular object. But it is not good to express a regular object. The regular object tetrahedral partition expression increases unnecessary calculations in some degrees. Considering the spatial regularity and its internal uniformity of solid elements, we divided solid objects into the following categories, and expressed them using different ways:

- Regular homogeneous objects: expressed using regular geometry primitives or their combinations;
- Irregular homogeneous objects: its whole solid could be represented by its surface. It could be expressed by the combinations of several face objects in order to simplify the raster procession. Surface is expressed using regular grids;
- Regular heterogeneous objects: expressed using regular geometry primitives and tetrahedrons;
- Irregular heterogeneous objects: expressed using tetrahedral partition.

Regular geometric primitives and tetrahedrons obviously become the basic geometric primitives to express solid objects. Therefore, it is necessary to establish a library for geometric primitives.

4.2 Establishment of Geometry Primitives Library

It is same as libraries in AutoCAD and 3DMAX, the following geometry primitives in our integrated data model are included in the geometric primitives library: Rectangle, Triangle, Cuboid, cylinder, cone, prism and sphere, Frustum of a pyramid, Tetrahedron. In addition, any geometry primitives that have analytical forms could be considered as regular primitives, and could be included in the geometric primitive library.

In order to express more complex objects, the library should be an open graphic library. It will allow users to add more geometric primitives in order to express some special objects. Besides, any new geometric primitive added to the library should be associated with an interpolation algorithm for vector and raster data integration. There should also be some basic operational functions, such as rotate, pan and zoom, in the library.

4.3 Raster interpolation Algorithm

In order to do the integration, vector elements have to be converted to raster cells. In our integrated data model, we will use raster interpolation methods with Code-Points. Now we discuss the interpolation methods for line, face and solid objects, respectively.

Line Object: a line object composes of a set of points. Assuming that two points are linked with a straight line, a line interpolation algorithm could be illustrated in Figure. 2, that is to calculate all raster cells along the line AB.

A vector line $L = \{p_{1(x,y,z)}, p_{2(x,y,z)}, \dots, p_{n(x,y,z)}\}$ in raster space should be $L = \{p_{1(\text{code})}, p_{2(\text{code})}, \dots, p_{n(\text{code})}\}$. The vector line L should be continuous when the length of Octree-ID is equal to 1. The deviation of $p_{1(\text{code})}, p_{2(\text{code})}, \dots, p_{n(\text{code})}$ is positive proportional to the length of Octree-ID.

We assume that any two Code-Points are connected with a straight line. The interpolation between two code-points is to calculate raster cells passed by the line segment. Each cell is expressed by an octal Morton code. Figure 2 shows, for a 2-dimensional plane, the four corners of the cell where Code-Point A is are A_1, A_2, A_3, A_4 , respectively. And the slopes of the four segments A_1A, A_2A, A_3A, A_4A are l_1, l_2, l_3, l_4 , respectively. The slope of the segment AB is l . Assume $x = B_x - A_x$ and $y = B_y - A_y$. Following the positive and negative values of x and

y , the next cell could be located by comparing the value between l and l_1, l_2, l_3, l_4 , respectively. For 3-Dimensional space, many more judgment conditions need to be added. This algorithm is similar to the linear scan conversion in computer graphics. A hardware chip could be designed to perform this process.

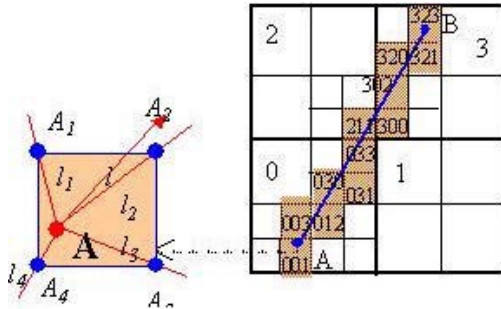


Fig. 2. Interpolation of a line using raster cells

Face object: In 2D plane, a face usually is composed of lines. The interpolation of a face could be processed on the base of linear interpolation algorithms. But in 3D space, a face object is usually very complex. For the convenience of raster interpolation, we only consider to express face objects using regular grid and TIN. Now, the question is the interpolation of spatial rectangles and spatial triangles.

The interpolation algorithm for rectangles could be considered in the way shown in Figure 3. A spatial rectangle has projections on three planes in 3D coordinate system respectively. The

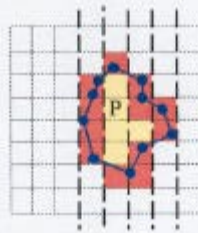


Fig. 3. Raster cell Interpolation of a polygon

projections are parallelograms on planes. For the interpolation algorithm for 2D plane area (see Fig. 3), it is simple to calculate the raster cell sets in three directions of x , y and z . From the calculation of their intersections, we could get a raster cell set of the spatial rectangles. Repeating the same way to all spatial rectangles that are used to express the face objects, all raster cells of a face object could be calculated.

To write a 2D face object interpolation algorithm, we assume that one polygon is surround by n lines. Because the polygon on the plane is a close area, there are at least two same X direction Code-Points (or X values) in Y direction. All other cells in Y direction between two cells must belong to the polygon. From above algorithm, we could find out all raster cells of the polygon.

```

Algorithm_VtoR_Poly(P)
{ // Let a polygon is P = {L1, L2,...,Ln};
  // All raster cells are stored in the table "Poly_Cell_List"
  for(i = 0; i < n; i++)
  {
    Algorithm_VtoR_Line(Li);
    Add_Line_Cell_To_Poly_Cell_List( );
    Temp_cell_List = Poly_cell_List;
    While(! IsEmpty(Temp_Cell_List) )
    {
      if( Find_Cell_Couple( )= TRUE)
        Find_Cell_Between_the_Couple();
      Add_cell(Poly_cell_List);
      Remove_the_Cell( Temp_Cell_List);
    }
  }
}

```

Solid Object: Comparing with face objects, solid objects are much more complex. Firstly, a solid object has to be split according to the elements in the geometric primitive library. Each primitive need to be processed with corresponding algorithms. For large primitive sets, the special high efficient algorithms for each primitive have to be used, especially for the operation algorithms of these primitives, such as rotate, pan and zoom etc. As our aim is the real-time raster interpolation for solid objects, the algorithms for every geometry primitives could not be too difficult to run in real time.

For geometric primitives expressed analytically, such as cuboids, cylinders and cones, etc., the centerlines or center points could be used for calculation. Frustums of a quadrangle or a polygon may be partitioned into tetrahedrons, and then processed using tetrahedrons. Frustum of a right circular cone could be calculated analytically as well. The raster cell interpolation for irregular and heterogeneous solid is possibly the most difficult process. Sometimes, these are hundreds of tetrahedrons are needed to express those objects. If a large number of such objects have to be processed at the same time, the real-time interpolation becomes much more difficult. In order to solve this problem, parallel algorithms are needed by grouping the tetrahedrons according to their Octree_ID. If a parallel process has ten thread processes, all tetrahedrons of the object can be divided into ten groups. Delete repeat cells in the results, we can get all raster cells of the object.

In order to process raster interpolation much fast, the whole geometric primitives library can be hardware realized, including those interpolation algorithms of each

primitives. We call this hardware “raster interpolation card”. The vector solid object can be split using primitives, and can be grouped according to their types. Tetrahedrons have to be grouped separately because there are usually a large number of them. All grouped data will be sent into “raster interpolation card” for processing. There are many calculation cells for different primitives types on cards. These cells are parallel to each other. The repeat raster cells have to be removed from results. The rest of raster cells will be integrated before they are used to express raster solid objects and face objects.

4.4 Integrated Data Model

From above analysis, we could conclude an integrated data model (Figure 4). The data model includes a raster space and a vector space at the same time. Objects are expressed using raster cell sets in the raster space and expressed as same as traditional methods in the vector space. A geometric primitive library was established. Face objects and solid objects are expressed using geometric primitives from the library. Polygon is introduced as a primitive for face objects in 2-Dimensional plane. Point elements are converted from vector to raster space using spatial code-points. But lines are converted using linear interpolations and face and solid objects are converted using the geometric primitive library.

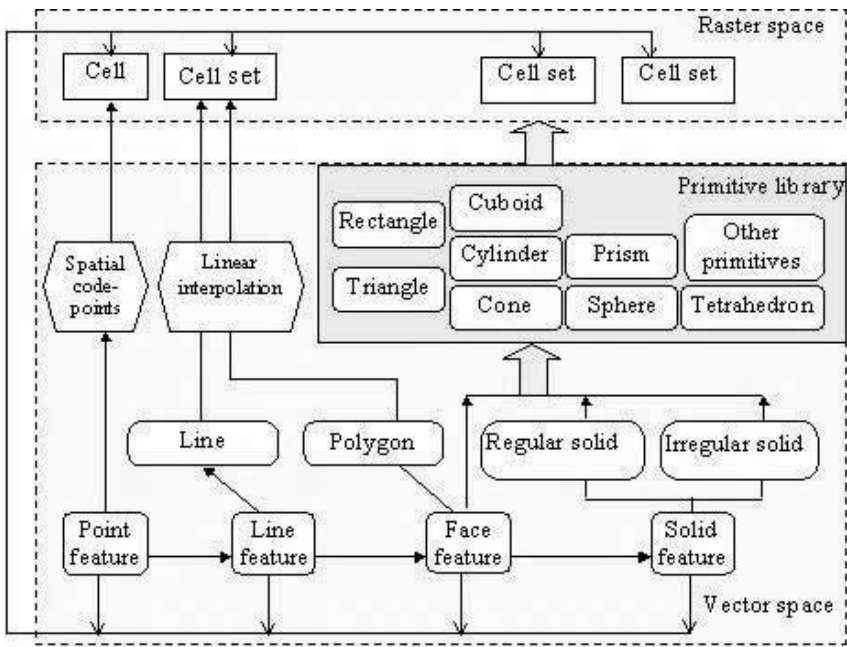


Fig. 4. Diagram of Raster and vector integrated data model

5. Data Structure

The foundations of the integrated data model are Code-Points and raster interpolation algorithms for vector objects. Although we have encapsulated geometric primitives that are used to express face objects and solid objects into a primitive library, the data model does not show object organizations. Hence, we need to discuss the data structure of the data model.

Table structure is the popular data structure in traditional 2-Dimensional GIS. The utilization of Oracle's Spatial Cartage module in GIS realizes the integrated management of vector spatial data and attribute database on RDBMS. Spatial Cartage uses quardtree index to manage spatial objects. In our integrated data model, octree is used to divide 3-dimensional space. Octree-IDs are used to manage vector points. But for other vector elements, such lines, face and solid elements, it is difficult to manage them. We develop a data structure based on list structure to manage those vector elements.

In order to manage point elements, we define a structure *Point_Code* {**point*, *Octree-id*} and use a list to manage it. Structure *Point_Code* is sorted according to code-point values in the list. As it's easy to insert, delete and find in the list, this structure is efficient to manage point objects. For line, face and solid objects, each object may have a large amount of space code-points. It cannot be organized same as the methods for point objects. However, these Octree-IDs for code-points could show the spatial positions of the objects. For example, a line L is composed of a set of points. These points' Octree-id are {543001 543021 543022 543123, ..., 543701}. We could use 543xxx to roughly index the line L in a raster space. The same way could be used to index face and solid objects. For face objects that are composed of lines and solid objects that are composed of faces, their line and face space code-points have to be integrated respectively). When a software system is in an interactive graphic mode, visible part is usually in local. So it is easy to do dynamic Octree-id integration in real-time.

6. Conclusion

Integrated data model is very useful for the integration of remote sensing and GIS. It is also a very difficult issue. Especially it is difficult to establish a simple and feasible data model. In this paper we have developed some methodologies. The advantages of our integrated data model are: (1) Partly pre-process of the data and real-time raster interpolation algorithms not only increase the flexible of the system, but also avoid the data redundancy of storages for raster data although Code-Point does not solve the problem of management of line, face and solid objects completely; (2) Conversion of vector data to raster data is usually a one-off process. In this data model, the process comes to be pretreatment (e.g. to find spatial code-points) and processing using algorithms in the local area. And it also could be processed in parallel based on Octree-ids; (3) It could be used to manage spatial objects efficiently although the index based on space code Octree-id is not as easy as octree index. It avoids the infeasibility caused by the excess occupation of RAM space from octree structure

itself; (4) The data model keeps the original vector characteristics, such as high precision and fewer amounts of data. And it also does not increase the complex of object expression and data operation. In addition, this integrated data model is compatible to existing vector data models in 3-Dimensional GIS systems.

Topological relationships are not included in this data model. Although expression and procession of spatial topological relationships distinguish GIS from other graphics processing systems especially CAD system, there is no big difference between GIS and CAD when the expression forms turn to be processed by algorithms. The question becomes whether we must express topological relationships. In fact, very limited topological relationships can be expressed in current 2-Dimensional and 3-Dimensional GIS data models. It is obviously a very difficult task if we try to calculate spatial relationships for real spatial objects from several simple topological relationships expressed in data models. Further more, expressions of topological relationships usually limit the flexibility of the data model and introduce the complexity for data updating. We propose that topological relationships should also be calculated in real-time.

Acknowledgments

This publication is an output from the research projects "CAS Hundred Talents Program", "Initial Phase of the Knowledge Innovation Program" and "Digital Earth" funded by Chinese Academy of Sciences (CAS) and is partly supported by a grant from the Chinese National Sciences Foundations (No. 59904001).

References

1. Ballard, D. H.: Strip Trees Hierarchical Representation for Curves, *CACM*, 24 (1981) 310-321.
2. Gong, J. Y.: 1993, The Research on Integrated Spatial Data Structure. Wuhan Technical University of Surveying and Mapping, PhD thesis, 1993 (In Chinese)
3. Li, R. X.: Data Structure and Application Issues in 3-D Geographical Information Systems. *Geomatics*, 48 (1994) 209-224.
4. Molenaar, M.: A Formal Data Structure For Three Dimensional Vector Maps, Proceedings of the 4th International Symposium on Spatial Data Handling, Zurich, 1990, pp830-843.
5. Pilouk M., Tempfli K., Molenaar M.: A Tetrahedron-Based 3D Vector Data Model for Geoinformation, *Advanced Geographic Data Modeling*. Netherlands Geodetic Commission, Publications on Geodesy, 40 (1994) 129-140.
6. Randal C., Samet H.: A Consistent Hierarchical Representation for Vector Data, *Siggraph*, 20 (1986) 197-206.
7. Sun, J.G., and Yang, C.G.: 1997, *Computer Graphics* (Tsinghua University Press, Beijing) (In Chinese)
8. Tempfle, K.: 3D topographic mapping for urban GIS. *ITC Journal*, 3/4 (1998) 181-190.
9. Wilkinson, G. G.: A review of current issues in the integration of GIS and remote sensing data. *International Journal of Geographic Information System*, 10 (1996) 85-101.